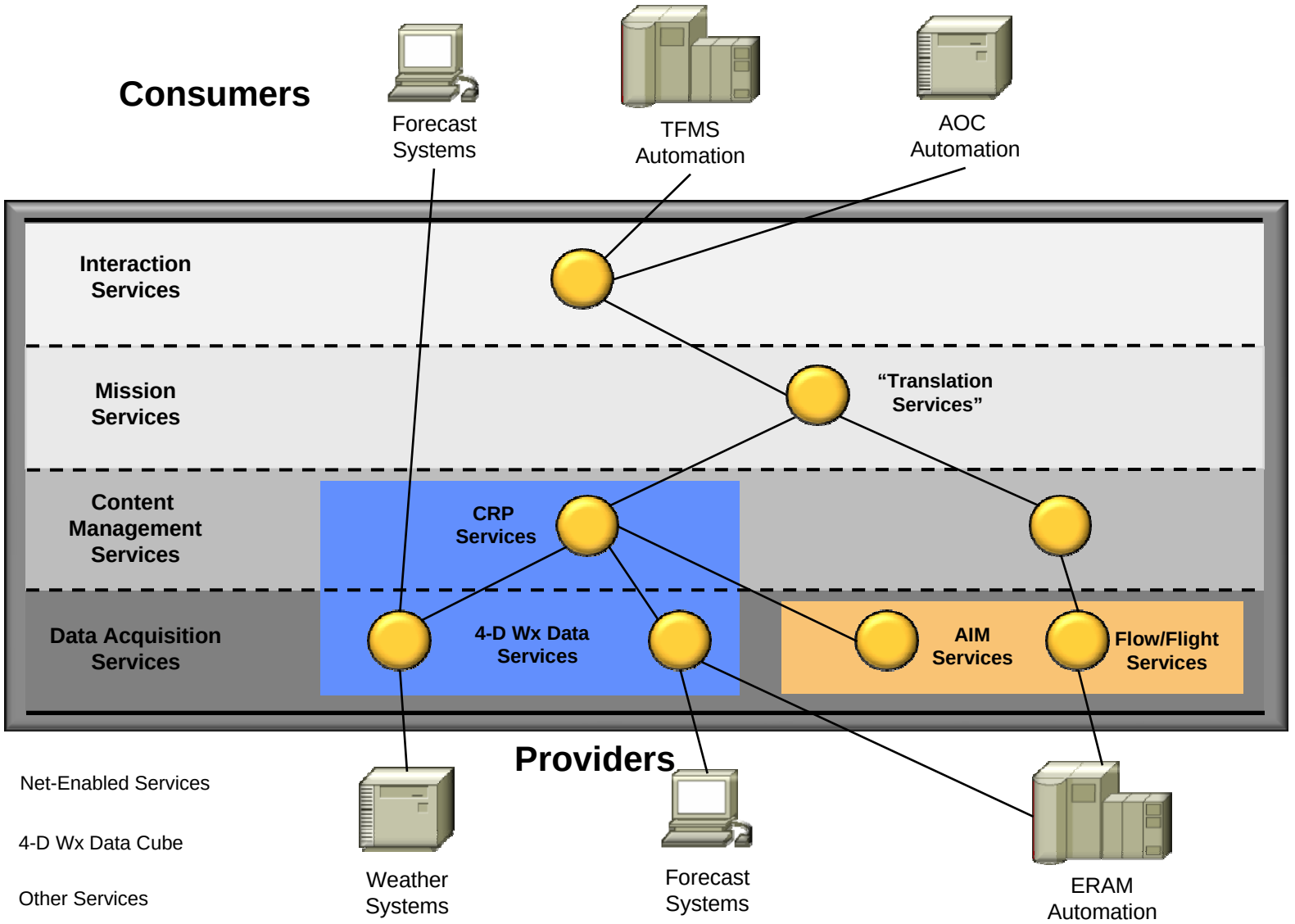
Disseminating WXXM Data via A Web Feature Service

Dr. Kajal T. Claypool

AIXM / WXXM 2010 Conference

05 May 2010

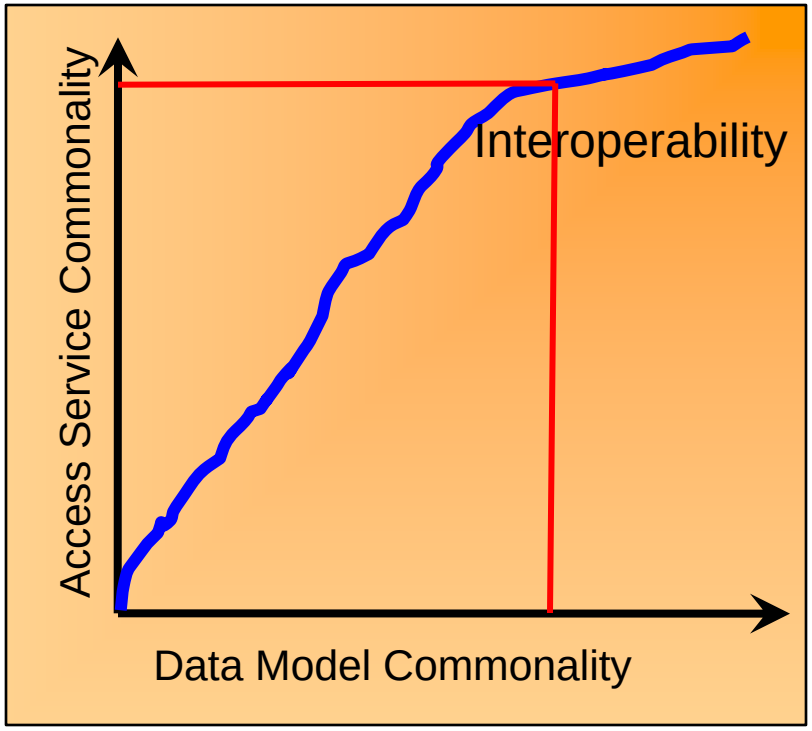
4-D Wx Data Cube SOA Context



“The Vision”

- ***Geographic Information Interoperability***
  - **Key Ingredient: Standardization!**

*



- Data Model: how is domain represented ?
  - common (standards!) foundation aids interoperability
  - *think: GML. ISO standards*
- **Services: how is “information” accessed?**
  - standard access methods aid interoperability
  - *think: OGC service standards*

*** Holds within and across domains!*

* Non-scientific co-relation

OGC Service Standards

- The OGC Web Coverage Service and **Web Feature Service** are *data access standards.*
- Oriented towards filtered request/response patterns, such as geographic filtering. These specifications define a general-purpose language for data clients to request data of interest
- Applicable outside of the weather domain (AIM WFS)
- These specifications are a (service) component of a system, and do not themselves solve architecture or specify a complete data dissemination system

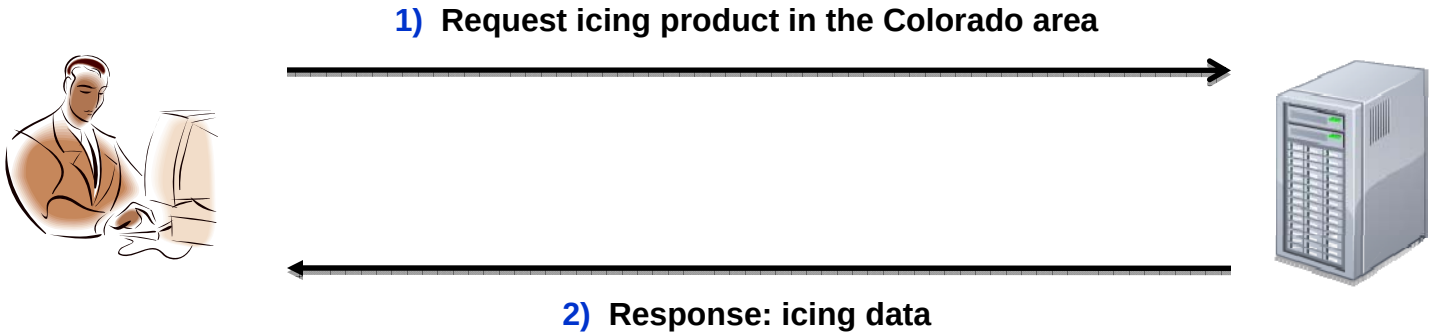
Outline

- Motivation
- NNEW Requirements
- NNEW WFS Reference Implementation
- Preliminary Performance Results
- Conclusions

Message Exchange Pattern: Request/Response

5.1.3.3 Retrieve icing forecast product for Colorado

Retrieve (via pull mechanism) icing potential forecast product for the geometry of Colorado for 2007-10-12 12:00Z (future time, 12 hours from the time the request is made), from 5000 ft to 35000 ft

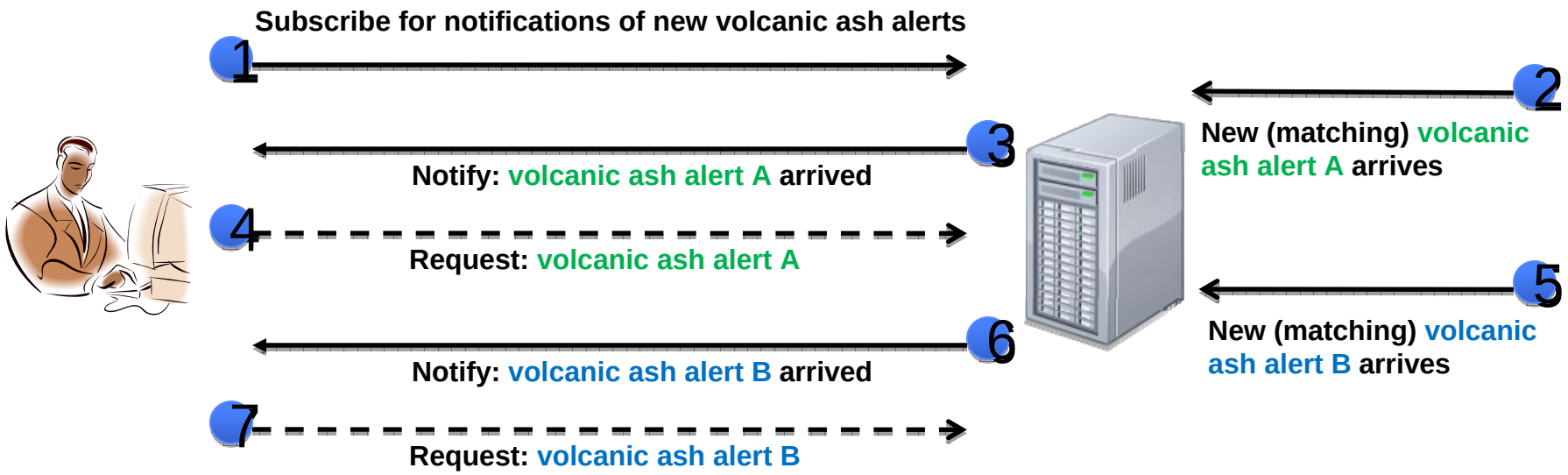


- **Synchronous sequence**
- **User must ask (poll) for new information**

Message Exchange Pattern: Notification

5.1.3.8 Subscribe to volcanic ash alerts for a specified flight path

Retrieve notification of volcanic ash (AIR/SIGMET) that is within 200 statute miles of an indicated flight path. The flight is planned for 1300Z on the 12th day of the month and notification is requested for the time period of 0Z on the 10th to 1800Z of the 12th.

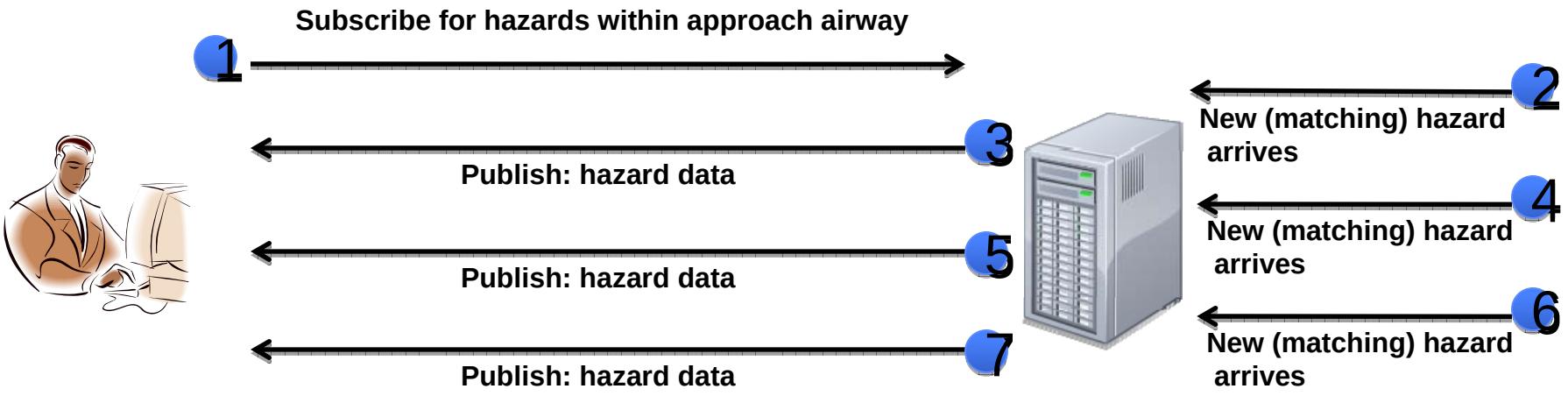


- Notifications pushed to data consumer
- Asynchronous sequence
- Persistent subscription
- No data is passed, only notifications of data arrival

Message Exchange Pattern: Persistent Query

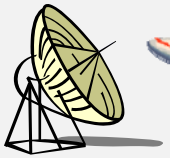
5.1.3.9 Monitor hazards in the terminal approach airway

A TRACON air-traffic controller needs to monitor adverse weather conditions that exist, or are forecasted to exist, within the approach airway. The approach airway is defined as a volume around an airport, defined by a set of XYZ vertices, within which departing and landing flights must fly.



- Data is pushed to data consumer
- Asynchronous sequence
- Persistent subscription

WFS Information Management



Controllers, CIWS, ITWS, ATC personnel, etc

Publishers insert data incrementally as XML documents. Insertions are validated against schema definitions.

Consumers receive filtered XML-formatted data via *persistent query* subscription mechanism.

Pub/Sub-enabled WFS Server

NNEW SOA implemented as Open Geospatial Consortium's (OGC) Web Feature Service (WFS)

Data automatically aged out of real-time database, into data warehouse

Data Store

Long-term data store

NNEW needs to provide a single, flexible service (OGC's WFS*) to other entities in the System-Wide Information Management (SWIM) SOA

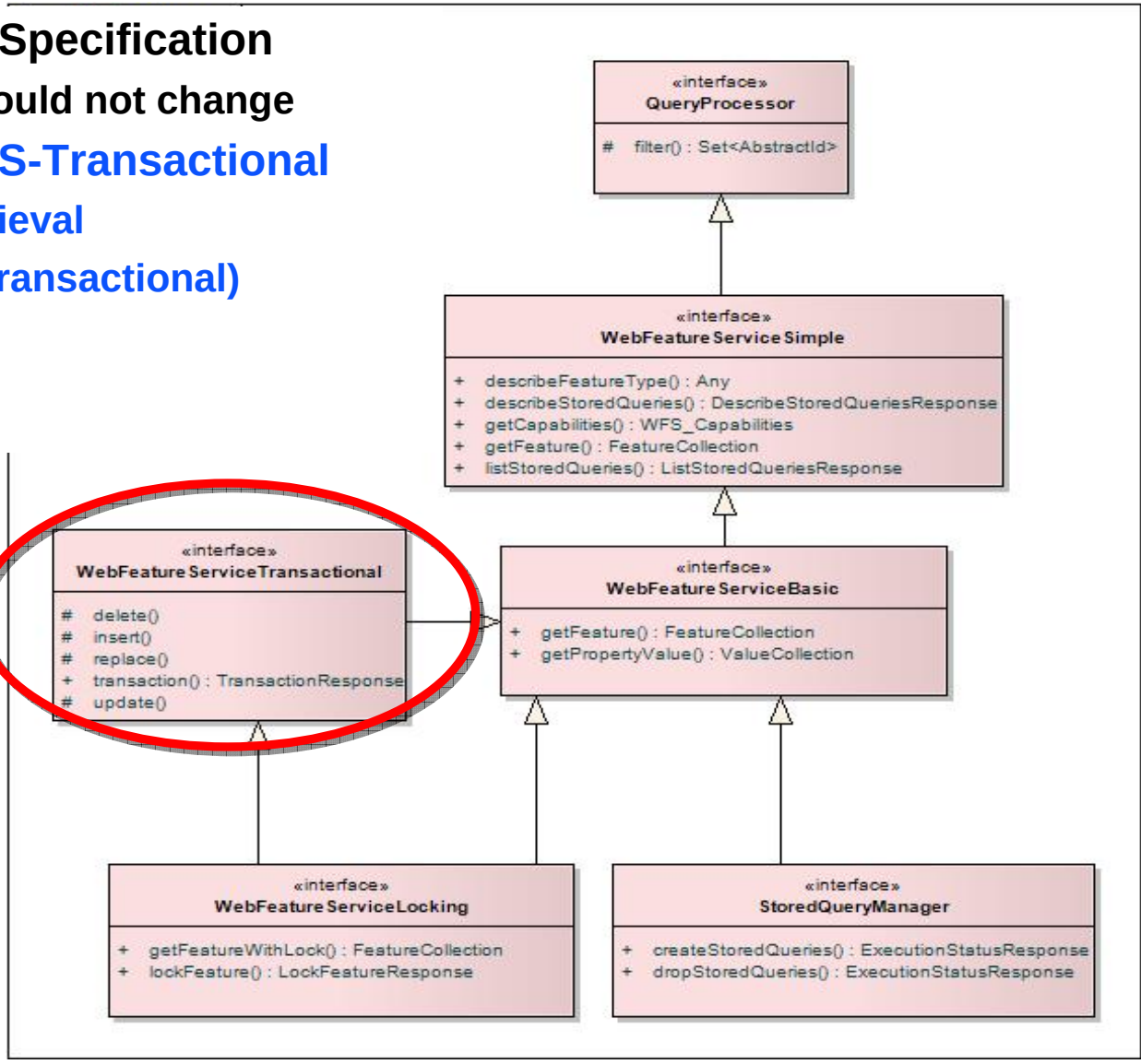
*The Open Geospatial Consortium's Web Feature Service

Outline

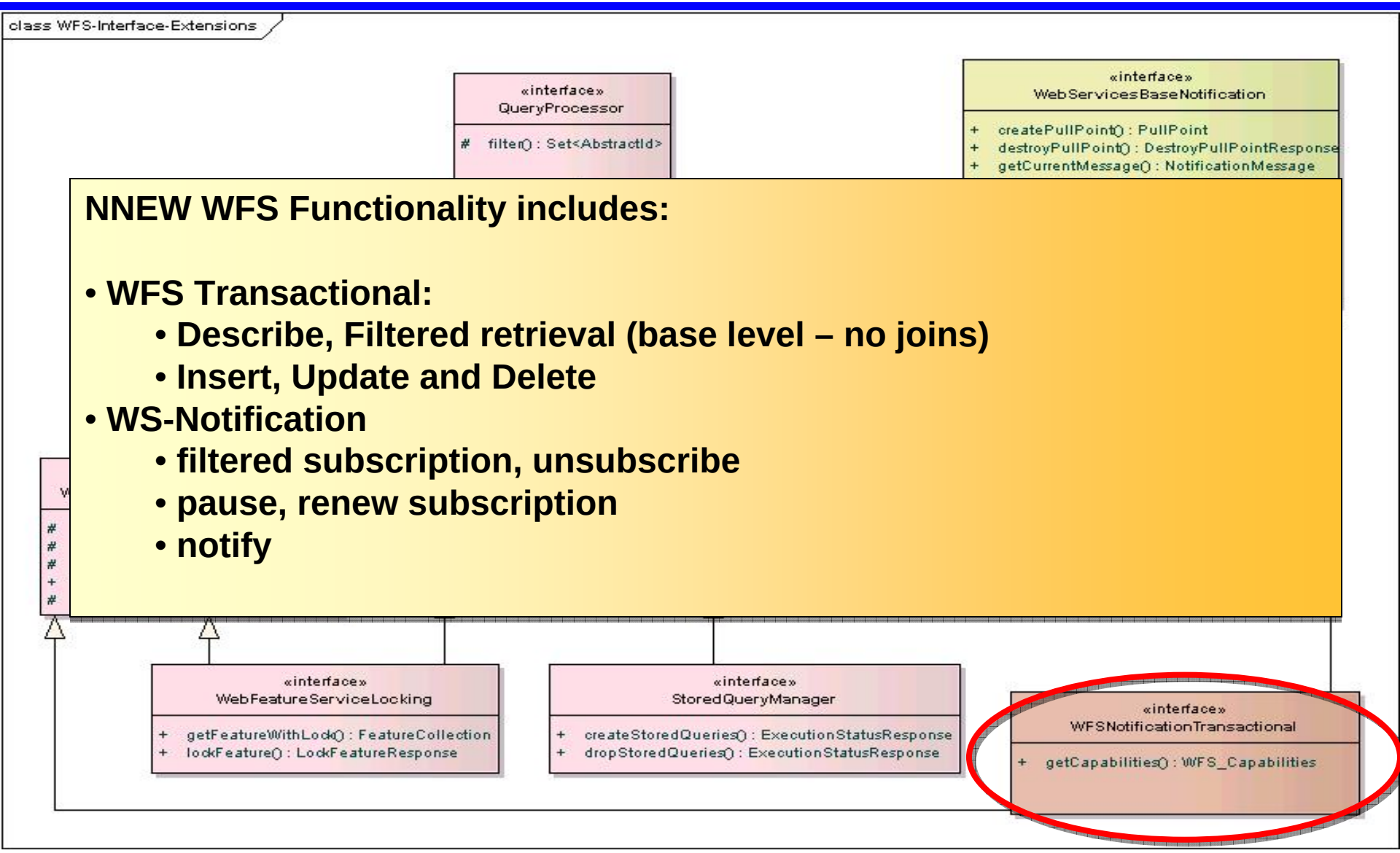
- Motivation
- NNEW Requirements
- NNEW WFS Reference Implementation
- Preliminary Performance Results
- Conclusions

OGC Web Feature Service (WFS)

- **OGC/ISO TC 211 WFS 2.0 Specification**
  - **Currently a draft but should not change**
- **NNEW WFS supports: WFS-Transactional**
  - **Discovery, Filtered Retrieval**
  - **Insert, Update, Delete (transactional)**
- **GML 3.2**
  - **WXXM 1.1**



NNEW WFS

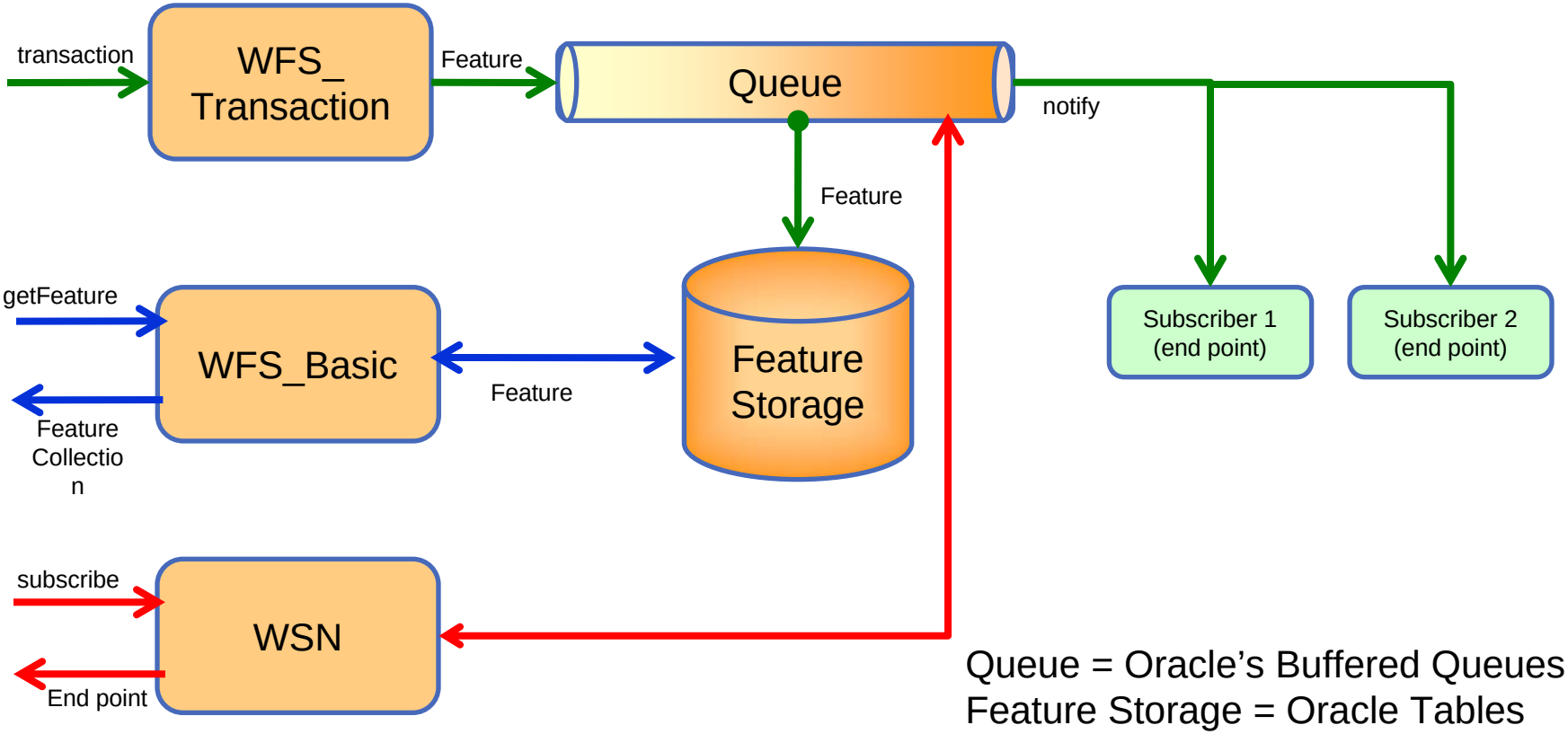


NNEW WFS Functionality includes:

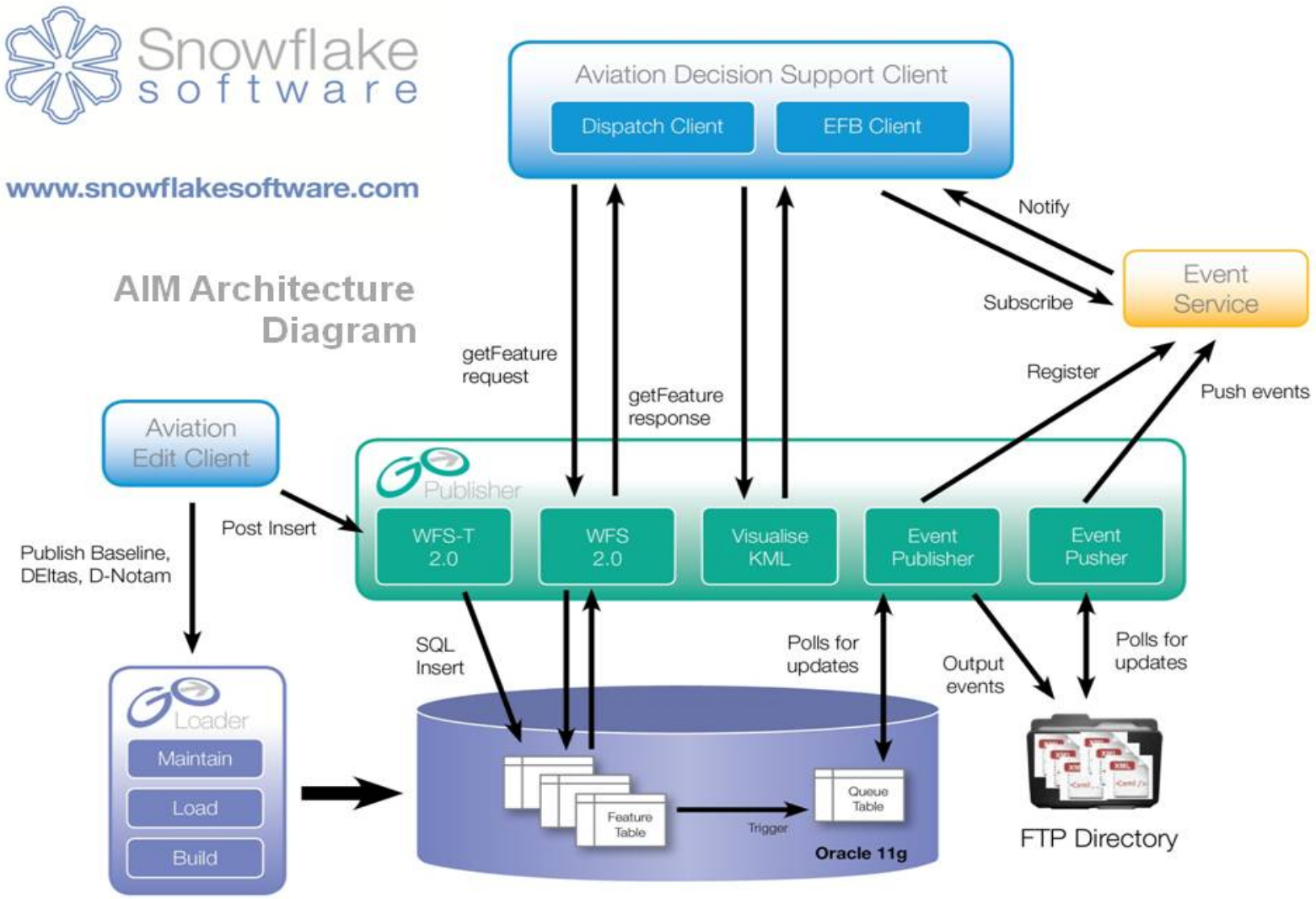
- **WFS Transactional:**
  - **Describe, Filtered retrieval (base level – no joins)**
  - **Insert, Update and Delete**
- **WS-Notification**
  - **filtered subscription, unsubscribe**
  - **pause, renew subscription**
  - **notify**

NNEW WFS : Wire Tap Pattern

- **Uses the Enterprise Integration Wire Tap design pattern**



Snowflake WFS Architecture



* Courtesy of Snowflake Software

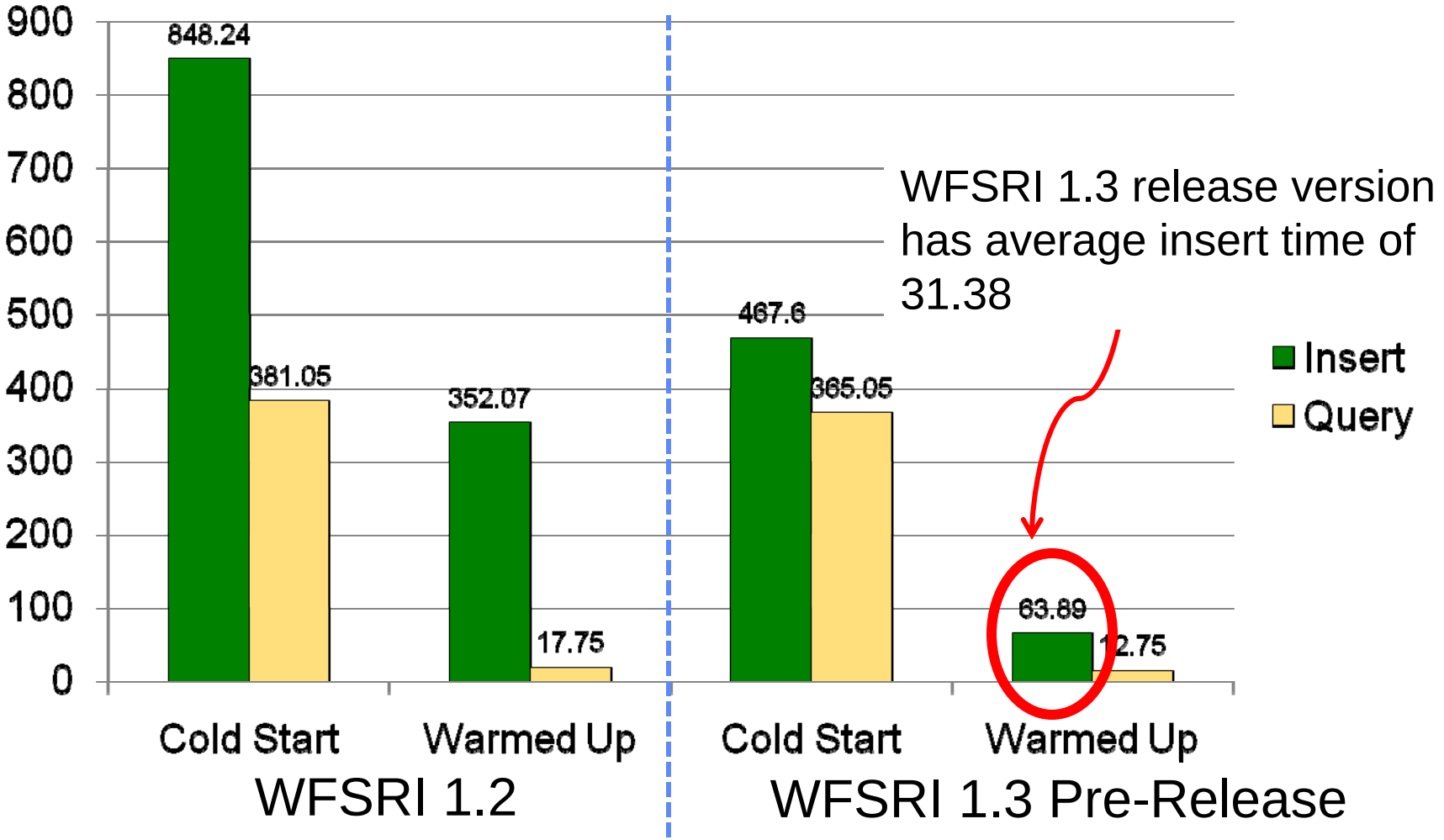
Outline

- Motivation
- NNEW Requirements
- NNEW WFS Reference Implementation
- Preliminary Performance Results
- Conclusions

Setup

ORACLE Server	WFSRI Server
Dell Power Edge R710	Dell Power Edge R410
Intel XEON E5550 (2) Quad-Core CPUs at 2.6GHz	Intel XEON E5520 (2) Quad-Core CPUs at 2.26GHz
16 GB (8X2GB) at 1333 MHz, UDIMM	16 GB (4X4GB) at 1066 MHz, RDIMM
(2) 1TB, 7.2K RPM H SATA, RAID-1	(2) 250 GB, 7.2K RPM HD SATA
Linux CentOS 5.4 with Kernel 2.6.18	Linux Cent OS 5.4 with Kernel 2.6.18
Oracle 11g Release2, 64 bit	WFSRI 1.3 (pre-release, svn trunk version 706).
JDK 1.6.0	JDK 1.6.0
Apache Tomcat 6.0.26, Apache Active MQ 5.3.1, Apache Ant 1.8.0.	Apache Tomcat 6.0.26, Apache Active MQ 5.3.1, Apache Ant 1.8.0.
1G bps Broadcom 5709 Ethernet	1G bps Broadcom 5709 Ethernet

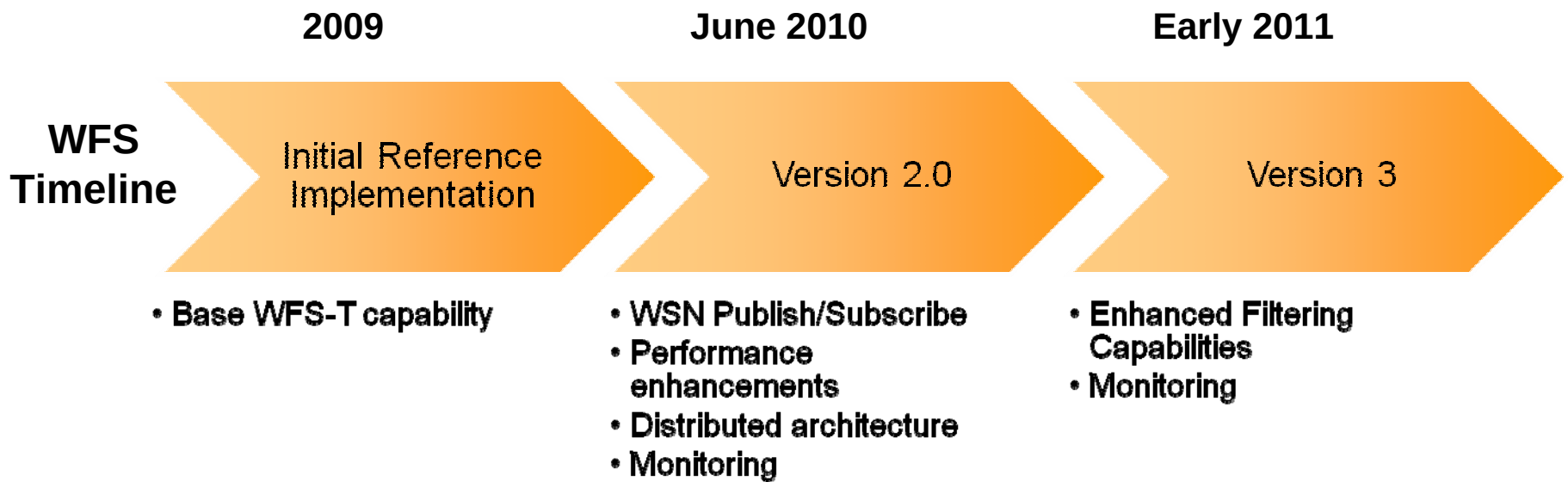
Mean Execution Time



* Experiments conducted by WPI

Next Steps for the RI

- **Optimization:**
  - would like to support 200 inserts/second
- **Distributed WFS:**
  - Support NNEW distribution scenario
  - Improved scalability
- **“Baby” Steps towards Complex Retrieval functionality**
  - Geospatial and Temporal joins



Acknowledgements

Many thanks to

Ai-Hoa Sanh
Kelly Moran
Aaron Braeckel (nice slides. Thanks!)
Oliver Newell
Dan Tennant

NNEW Team

WPI

Debbie Wilson (Snowflake Software)

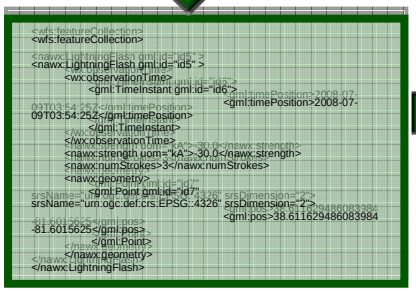
&

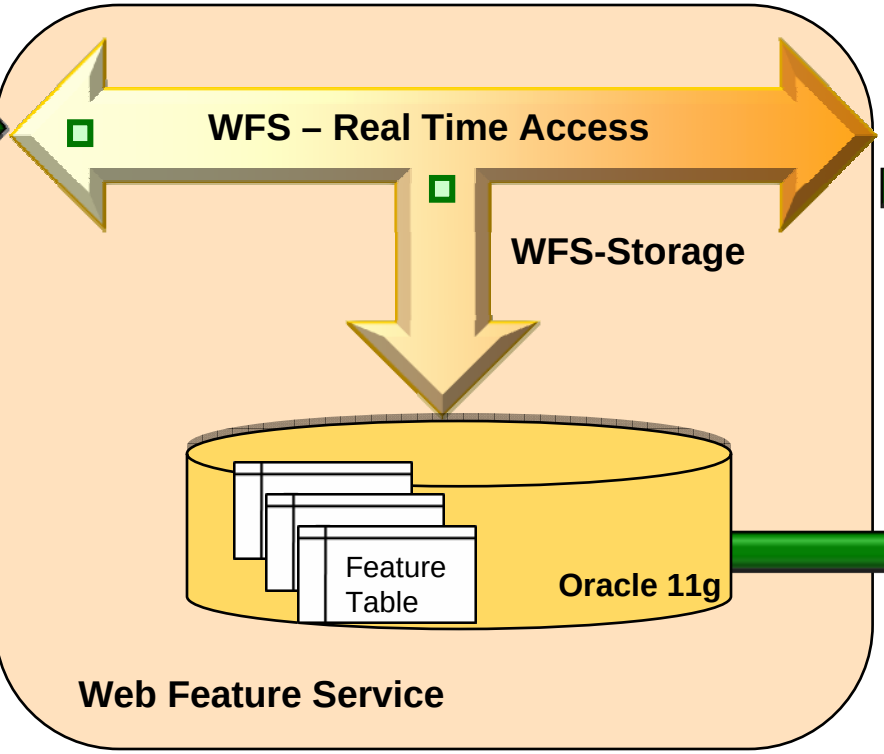
All our "users" – Thank you for finding the bugs!

Questions:
Kajal Claypool (claypool@ll.mit.edu)

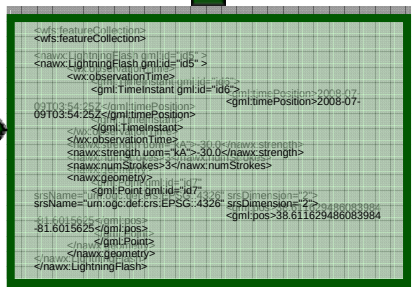
EIP: Wire Tap Pattern

Data Publisher





Real Time Client



Asynchronous Access (Request/Response)

WFS Architectural Goals

- Robust and dependable reference implementation
- Fault tolerant to network and machine failure
  - Must support graceful recovery.
  - Must not result in data corruption or security malfunction.
- Scalable to multiple nodes
  - Primarily LAN-based, with limited WAN possibilities
- Performance
  - Should provide fast response time (on the order of milliseconds/seconds) once a user request has been issued.
  - Must be able to handle possibly concurrent requests.
- Modifiable and extensible implementation to support future needs
- Out of box solution